Journal of Computational Physics 228 (2009) 96–108

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

The multidimensional moment-constrained maximum entropy problem: A BFGS algorithm with constraint scaling

Rafail V. Abramov *

Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, 851 S. Morgan Street, Chicago, IL 60607, United States

ARTICLE INFO

Article history:
Received 8 June 2008
Accepted 26 August 2008
Available online 5 September 2008

Keywords:
Maximum entropy problem
Moment constraints
Convex optimization
Numerical methods

ABSTRACT

In a recent paper we developed a new algorithm for the moment-constrained maximum entropy problem in a multidimensional setting, using a multidimensional orthogonal polynomial basis in the dual space of Lagrange multipliers to achieve numerical stability and rapid convergence of the Newton iterations. Here we introduce two new improvements for the existing algorithm, adding significant computational speedup in situations with many moment constraints, where the original algorithm is known to converge slowly. The first improvement is the use of the BFGS iterations to progress between successive polynomial reorthogonalizations rather than single Newton steps, typically reducing the total number of computationally expensive polynomial reorthogonalizations for the same maximum entropy problem. The second improvement is a constraint rescaling, aimed to reduce relative difference in the order of magnitude between different moment constraints, improving numerical stability of iterations due to reduced sensitivity of different constraints to changes in Lagrange multipliers. We observe that these two improvements can yield an average wall clock time speedup of 5–6 times compared to the original algorithm.

1. Introduction

The moment-constrained maximum entropy problem yields an estimate of a probability density with highest uncertainty among all densities satisfying supplied moment constraints. The moment-constrained maximum entropy problem arises in a variety of settings in solid state physics [5,16,17,22], econometrics [20,23], and geophysical applications such as weather and climate prediction [3,4,15]. The approximation itself is obtained by maximizing the Shannon entropy under the constraints established by measured moments (phase space-averaged monomials of problem variables).

Recently the author developed new algorithms for the multidimensional moment-constrained maximum entropy problem [1,2]. While the method in [1] is somewhat primitive and is only capable of solving two-dimensional maximum entropy problems with moments of order up to 4, the improved algorithm in [2] uses a suitable orthonormal polynomial basis in the space of Lagrange multipliers to improve convergence of its iterative optimization process, and is practically capable of solving two-dimensional problems with moments of order up to 8, three-dimensional problems with moments of order up to 6, and four-dimensional problems of order up to 4, totalling 44, 83 and 69 moment constraints, respectively.

The algorithm in [2] demonstrated good numerical stability and reliable convergence for a variety of maximum entropy problems with different moment constraints, including probability density functions (PDFs) with complex shapes. It was not designed to achieve high computational speed, since at that point numerical stability and the ability to converge for a multidimensional problem with many constraints and a complex target PDF were the main goals of the algorithm development.

* Tel.: +1 312 413 7945; fax: +1 312 996 1491.
E-mail address: abramov@math.uic.edu

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.08.020

However, at the current stage it became clear that the algorithm, developed in [2], is stable and robust under a variety of multidimensional maximum entropy problems, but can still be improved to achieve higher computational speeds while retaining same numerical stability and reliable convergence for complex multidimensional maximum entropy problems.

In the current work we develop and test a new improved algorithm for the multidimensional moment-constrained maximum entropy problem, which utilizes multiple Broyden–Fletcher–Goldfarb–Shanno (BFGS) iterations [6,7,9,13,21] to adaptively progress between points of polynomial reorthonormalization, as opposed to single Newton steps introduced in [2]. This strategy aims to reduce the total number of computationally expensive polynomial reorthonormalizations needed to achieve the optimal point, thus typically resulting in reduced computational time for the same problem. Additionally, a new constraint rescaling is introduced to improve convergence by adjusting magnitude of the highest-order corner moment constraints to the low-order constraints, thus reducing negative effects of the finite-precision machine arithmetic in the iterative convergence process. As we demonstrate below, combining these two improvements can significantly reduce computational wall clock time for the same maximum entropy problem.

The manuscript is organized as follows. Section 2 outlines the basic algorithm with orthonormal polynomial basis developed in [2]. In Section 3 we introduce the following two improvements to the basic algorithm: the BFGS modification and the new constraint rescaling. Section 4 contains the results of tests for the two-, three- and four-dimensional setting with moments of order up to 8, 6 and 4, respectively. Section 5 summarizes the results of this work.

2. The basic method with a polynomial basis

We here start with formulation of the moment-constrained maximum entropy problem in a multidimensional domain. Here we follow notations introduced earlier in [1,2], where for $\vec{x} \in \mathbb{R}^N$, with N being the dimension of the domain, an arbitrary monomial of $\vec{x}$ (the product of arbitrary powers of components of $\vec{x}$) is concisely written as

$$\vec{x}^{\vec{i}} = \prod_{k=1}^{N} x_k^{i_k}, \qquad \vec{i} \in \mathbb{I}^N, \tag{1}$$

such that the monomial order $|\,\vec{i}\,|$ is the total power of all vector components, i.e.

$$|\vec{i}| = \sum_{k=1}^{N} i_k. \tag{2}$$

Using the above notation, for a probability density ρ we write the set of moment constraints of the total power up to M as

$$\mu_{\vec{i}}(\rho) = \int_{\mathbb{R}^N} \vec{x}^{\vec{i}} \rho(\vec{x})\, \mathrm{d}\vec{x} = m_{\vec{i}}, \qquad 0 \leqslant |\vec{i}| \leqslant M, \tag{3}$$

where m_0 is usually set to 1 to satisfy the normalization requirement for a probability distribution. Then, the solution to the maximum entropy problem is a probability density ρ^* which maximizes the Shannon entropy

$$S(\rho) = -\int_{\mathbb{R}^N} \rho(\vec{x}) \ln \rho(\vec{x})\, \mathrm{d}\vec{x} \tag{4}$$

(i.e. such that $S(\rho^*) = \max_\rho S(\rho)$) over all probability densities ρ which satisfy the moment constraints in (3).

The constrained entropy maximization problem can be reduced [1,2,24] to the unconstrained minimization of the Lagrangian function

$$\mathscr{L}(\rho) = S(\rho) + \sum_{|\vec{i}|=0}^{M} \lambda_{\vec{i}} (\mu_{\vec{i}}(\rho) - m_{\vec{i}}), \tag{5}$$

where $\lambda_{\vec{i}}$ are the Lagrange multipliers. A necessary condition $\delta\mathscr{L}/\delta\rho = 0$ for the minimum of (5) yields the form of the optimal probability distribution in terms of the Lagrange multipliers as

$$\rho^*(\vec{x}, \vec{\lambda}) = \exp\left(\sum_{|\vec{i}|=0}^{M} \lambda_{\vec{i}} \vec{x}^{\vec{i}} - 1 \right). \tag{6}$$

Note that, unlike in [1,2], here we treat the normalization constant λ_0 as a generic unknown Lagrange multiplier, while it is an explicit function of all other Lagrange multipliers. However, the current formulation simplifies the optimization problem below at the expense of one extra degree of freedom, and, as we observed empirically, makes the optimization path less "stiff" and generally improves convergence.

Substituting (6) into (5) and shifting the normalization constant λ_0 to $\lambda_0 + 1$ yields

$$\mathscr{L}(\lambda) = \int_{\mathbb{R}^N} \exp\left(\sum_{|\vec{i}|=0}^{M} \lambda_{\vec{i}} \vec{x}^{\vec{i}} \right) \mathrm{d}\vec{x} - \sum_{|\vec{i}|=0}^{M} \lambda_{\vec{i}} m_{\vec{i}}, \tag{7}$$

while the formula for the optimal probability density in (6) becomes

$$\rho^*(\vec{x},\vec{\lambda}) = \exp\left(\sum_{|\vec{i}|=0}^{M} \lambda_{\vec{i}} \vec{x}^{\vec{i}}\right). \tag{8}$$

The Lagrangian function in (7) is minimized over its Lagrange multipliers. The gradient and Hessian of (7) are, respectively,

$$(\nabla \mathscr{L})_{\vec{i}}(\vec{\lambda}) = \mu_{\vec{i}}(\rho^*) - m_{\vec{i}}, \tag{9a}$$

$$H_{\vec{i}\vec{j}}(\vec{\lambda}) = \mu_{\vec{i}+\vec{j}}(\rho^*). \tag{9b}$$

It is easy to show [1,2] that the Hessian in (9b) is a symmetric positive definite matrix, and, therefore, the Lagrangian minimization problem has a unique minimum, if exists. The minimum is reached when the gradient in (9a) is zero, which automatically satisfies the moment constraints in (3).

At this stage it is necessary to choose new coordinates for optimization, because the Lagrangian function (7) in the monomial basis $\vec{x}^{\vec{i}}$ has vastly different sensitivity to changes in different Lagrange multipliers, which negatively impacts convergence of standard iterative optimization methods. Following [2], we replace basis monomials $\vec{x}^{\vec{i}}$ with a set of Mth order polynomials $p_k(\vec{x})$,

$$\{\vec{x}^{\vec{i}}, \lambda_{\vec{i}}\},\ \vec{i} \in \mathbb{N}^N,\ 0 \leqslant |i| \leqslant M \to \{p_k(\vec{x}), \gamma_k\},\ 1 \leqslant k \leqslant K, K = \frac{(M+N)!}{M!N!}, \tag{10}$$

where γ_k are the Lagrange multipliers of the new basis. Since each basis polynomial $p_k(\vec{x})$ is of Mth order, the Lagrangian function should have comparable sensitivity to changes in different γ_k. The polynomials $p_k(\vec{x})$ are chosen at each step of iterative optimization to satisfy the orthogonality condition with respect to the current iterate of ρ^*

$$\int_{\mathbb{R}^N} p_k(\vec{x}) p_l(\vec{x}) \rho^*(\vec{x})\, d\vec{x} = \delta_{kl}, \tag{11}$$

which is done via the modified Gram–Schmidt algorithm [2,8,10–12]. The Lagrangian function (7) is written in the new polynomial coordinates as

$$\mathscr{L}(\vec{\gamma}) = \int_{\mathbb{R}^N} \exp\left(\sum_{k=1}^{K} \gamma_k p_k(\vec{x})\right) d\vec{x} - \sum_{k=1}^{K} \gamma_k p_k(\vec{m}), \tag{12}$$

where $p_k(\vec{m})$ above in (12) denotes the polynomial $p_k(\vec{x})$ where all monomials $\vec{x}^{\vec{i}}$ are replaced with corresponding values of constraints $m_{\vec{i}}$ from (3). The corresponding optimal probability density function is now written as

$$\rho^*(\vec{x},\vec{\gamma}) = \exp\left(\sum_{k=1}^{K} \gamma_k p_k(\vec{x})\right). \tag{13}$$

Due to the orthogonality requirement in (11), the Hessian matrix of (12) becomes the identity matrix

$$H_{kl}(\vec{\gamma}) = \delta_{kl}. \tag{14}$$

The minimization problem for the Lagrangian function in (12) is solved through the steepest descent iterative method, where the next iterate of the Lagrange multipliers is chosen as

$$\vec{\gamma}_{n+1} = -\alpha_n \nabla \mathscr{L}(\vec{\gamma}_n), \tag{15}$$

with α_n being the stepping distance determined by a standard inexact line search. With (14), the gradient descent in (15) coincides with the Newton algorithm and yields a second order convergence in the vicinity of the optimal point. The starting point for the iterations is chosen to be the Gaussian distribution with the mean and covariance matrices satisfying the constraints in (3) up to the second order. Standard shift, rotation and rescaling of coordinates [2] are used prior to the iterations, simplifying the optimization problem to the one with zero mean state and identity covariance matrix. A high-order Gauss–Hermite quadrature is used to compute the Gram–Schmidt orthonormalization and the gradient in (15). For more details on the preconditioning and quadratures see [1,2].

3. The new algorithm

The basic optimization algorithm with orthonormal polynomial basis, described above and in [2], is observed to have good numerical stability and reliable convergence for a variety of maximum entropy problems. Here we improve the basic algorithm to generally achieve higher computational speeds while retaining same numerical stability and reliable convergence for complex multidimensional maximum entropy problems, by using the BFGS iterations and the new constraint rescaling, described below.

3.1. The BFGS method

The basic optimization algorithm, described above in Section 2, performs the Gram–Schmidt reorthonormalization of the basis polynomials at each step of the steepest descent iterations in (15), which, on one hand, yields second order convergence in the vicinity of the optimal point, but, on the other hand, increases computational expense of the method, since the computational cost of polynomial reorthonormalization roughly matches the cost of computing Hessian in the monomial basis. Thus, an obvious way to reduce the computational expense of the basic algorithm is to perform several descent iterations between the Gram–Schmidt reorthonormalizations. However, the steepest descent method in (15) in this case loses precision down to the first order and becomes vulnerable to curvature anisotropy due to its lack of affine invariance. On the other hand, we need to avoid computing Hessian during the iterations between polynomial basis reorthonormalizations, which suggests using one of the quasi-Newton methods, such as the BFGS.

The Broyden–Fletcher–Goldfarb–Shanno formula [6,7,9,13,21] is a quasi-Newton method, widely used in optimization algorithms. It needs the Hessian (or an approximation to it) at the starting point of iterations and only requires computation of the gradient of the Lagrangian function in (12) at each successive iteration, which significantly reduces computational cost in our case. The structure of the BFGS algorithm is the following:

- In the beginning of iterations, provide starting gradient of the Lagrangian function $(\nabla\mathscr{L})_0$ and starting Hessian H_0, which is an identity matrix after polynomial reorthonormalization (see (14));
- At iteration m, perform the following steps:
  (a) Find the direction of descent by solving
  $$H_m\vec{d}_m = -(\nabla\mathscr{L})_m, \tag{16}$$
  (b) Perform a line search for the step distance α_m and find the next iterate $\vec{\gamma}_{m+1}$ as
  $$\vec{\gamma}_{m+1} = \vec{\gamma}_m + \alpha_m\vec{d}_m, \tag{17}$$
  (c) At the new iterate $\vec{\gamma}_{m+1}$ compute the gradient $(\nabla\mathscr{L})_{m+1}$,
  (d) Compute the new iterate of the pseudo-Hessian as
  $$H_{m+1} = H_m + \frac{\vec{y}_m\otimes\vec{y}_m}{\vec{s}_m\vec{y}_m} - \frac{(H_m\vec{s}_m)\otimes(H_m\vec{s}_m)}{\vec{s}_m H_m\vec{s}_m}, \tag{18}$$
  where $\vec{s}_m = \vec{\gamma}_{m+1} - \vec{\gamma}_m$ and $\vec{y}_m = (\nabla\mathscr{L})_{m+1} - (\nabla\mathscr{L})_m$.

In practice, to compute the descent direction in (16), we apply the Sherman–Morrison formula to the pseudo-Hessian in (18) and obtain

$$H^{-1}_{m+1} = H^{-1}_m + \frac{\vec{s}_m\vec{y}_m + \vec{y}_m H^{-1}_m\vec{y}_m}{(\vec{s}_m\vec{y}_m)^2}(\vec{s}_m\otimes\vec{s}_m) - \frac{1}{\vec{s}_m\vec{y}_m}[(H^{-1}_m\vec{y}_m)\otimes\vec{s}_m + \vec{s}_m\otimes(H^{-1}_m\vec{y}_m)]. \tag{19}$$

Then, the descent direction in (16) is computed as

$$\vec{d}_m = -H^{-1}_m(\nabla\mathscr{L})_m. \tag{20}$$

For details on the Sherman–Morrison formula see, for example, [14].

As successive BFGS steps change the current iterate of the probability distribution ρ^* in (13), the current set of polynomials p_k in (10), staying the same, gradually loses orthogonality with respect to changing ρ^*, which negatively impacts convergence of the BFGS iterations due to increased numerical errors in computation of the descent direction. Thus, when the error in the descent direction becomes too large, the basis polynomials in (10) have to be reorthonormalized with respect to the current iterate of (13) and the BFGS process has to be restarted. This adaptive reorthonormalization approach requires a computationally cheap estimate of the numerical error in descent direction to be available at each step of the BFGS iterations. One can observe from (20) that errors in the search direction $\vec{d}_m$ originate from the errors in the computed gradient of the Lagrangian function $\nabla\mathscr{L}$, amplified by the inverse pseudo-Hessian H^{-1}_m. The main source of errors in the gradient of the Lagrangian function are the Gauss–Hermite quadrature errors in computing moments of ρ^*, which usually remain bounded since the size of the quadrature and locations of abscissas are fixed during the course of computation. Thus, errors in the computed descent direction increase mainly when amplified by the inverse pseudo-Hessian H^{-1} in (20), and may grow significantly when H^{-1} becomes too ill-conditioned.

In the present algorithm, we monitor the condition number κ of the inverse pseudo-Hessian H^{-1} during the BFGS iterations and reorthonormalize the polynomial basis when κ exceeds the value of 20 (this threshold value of κ is empirically found to be small enough to preserve numerical stability of iterations, and at the same time large enough to allow multiple successive BFGS iterations between polynomial reorthonormalizations). Following [14], we compute the condition number in the L_∞ norm as

$$\kappa = \|H\|_\infty \|H^{-1}\|_\infty, \qquad \|H\|_\infty = \max_i \sum_j |H_{ij}|. \tag{21}$$

Thanks to the Sherman–Morrison formula, both H and H^{-1} are readily available through (18) and (19), respectively, and the computation of the condition number κ is inexpensive.

3.2. The new constraint scaling

In [1,2] we developed a preconditioning algorithm which rescales the supplied constraints to the new set of constraints which correspond to the target optimal probability density with zero mean state and identity covariance matrix. This preconditioning allowed to have a unique choice of the Gauss–Hermite quadrature scaling in all dimensions regardless of the initial constraints. However, one can observe that such a constraint scaling causes the moment constraints of high power to differ from the moment constraints of low power by several orders of magnitude, which may negatively impact convergence of the algorithm. As an example, let us look at the moments of a simple one-dimensional Gaussian distribution with zero mean state and unit variance. The explicit formula for this distribution is

$$\rho_G(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right). \tag{22}$$

The Gaussian distribution in (22) is an even function, and, therefore, all its moments of odd power are automatically zero. However, for the moments of (22) of even power it is easy to verify the following recursive relation

$$\mu_{M+2}(\rho_G) = (M+1)\mu_M(\rho_G), \qquad M \text{ even}, \tag{23}$$

and, taking into account the fact that $\mu_2(\rho_G) = 1$, we obtain the following formula for the even moments of the Gaussian distribution in (22)

$$\mu_M(\rho_G) = (M-1)!!, \qquad M \text{ even}, \tag{24}$$

where $(M-1)!!$ denotes the factorial over odd numbers up to $M-1$. As we can see, the Gaussian moments in (24) grow extremely fast with increasing M. This situation is further complicated by the fact that for a maximum entropy problem of moment constraints of order up to M the polynomial reorthonormalization procedure requires computation of moments of order up to $2M$. For instance, when the 8-order maximum entropy problem is started with the initial Gaussian distribution of zero mean and unit variance, the polynomial reorthogonalization algorithm computes $\mu_{16} = 15!! = 2,027,025$, which exceeds μ_0 and μ_2 (computed by the same algorithm) by 6 orders of magnitude.

To partially remedy the difference in magnitude between moments of different order, we suggest the following strategy. Instead of constraining the problem with the original set of moments, we will look for the optimal probability distribution

$$\rho^*_\alpha(\vec{x}) = \alpha^N \rho^*(\alpha\vec{x}), \tag{25}$$

with moment constraints

$$\mu_{\vec{i}}(\rho^*_\alpha) = \alpha^{-|\vec{i}|}\mu_{\vec{i}}(\rho^*), \qquad 0 \leqslant |\vec{i}| \leqslant M. \tag{26}$$

where α can be chosen to minimize difference in absolute magnitude between different moments. Note that for polynomial reorthonormalization the moments of total power up to $2M$ have to be computed, and therefore need to be taken into account for determining α.

A general way to address α is to set it to an $N \times N$ matrix, and then choose its entries at each polynomial reorthonormalization so that the relative distance between magnitudes of all the moments of order up to $2M$ is minimized, with subsequent rescaling of moment constraints. While such an elaborate strategy for α will, probably, be addressed by the author in the future, it is somewhat sophisticated and requires a separate optimization subproblem to be solved for α at each polynomial reorthonormalization, which could drive the computational cost of the problem further up. Instead, in the current work we choose to pursue a simpler strategy for α.

Here the scaling α is chosen once in the beginning of the optimization process. As the moments of the target probability distribution of order greater than M are unknown at the start, instead we choose α to set the values of the highest corner moments μ^n_{2M} to the value of the normalization constant μ_0 of the starting guess of the iterations. The corner moments μ^n_m are defined as follows

$$\mu^n_m(\rho) = \int_{\mathbb{R}^N} x_n^m \rho(\vec{x})\, \mathrm{d}x. \tag{27}$$

With the Gaussian starting distribution of zero mean and identity covariance matrix, all the corner moments of order $2M$ are equal to $(2M-1)!!$, while the normalization constant $\mu_0 = 1$. With this, α is taken to be a scalar

$$\alpha = \sqrt[2M]{(2M-1)!!}. \tag{28}$$

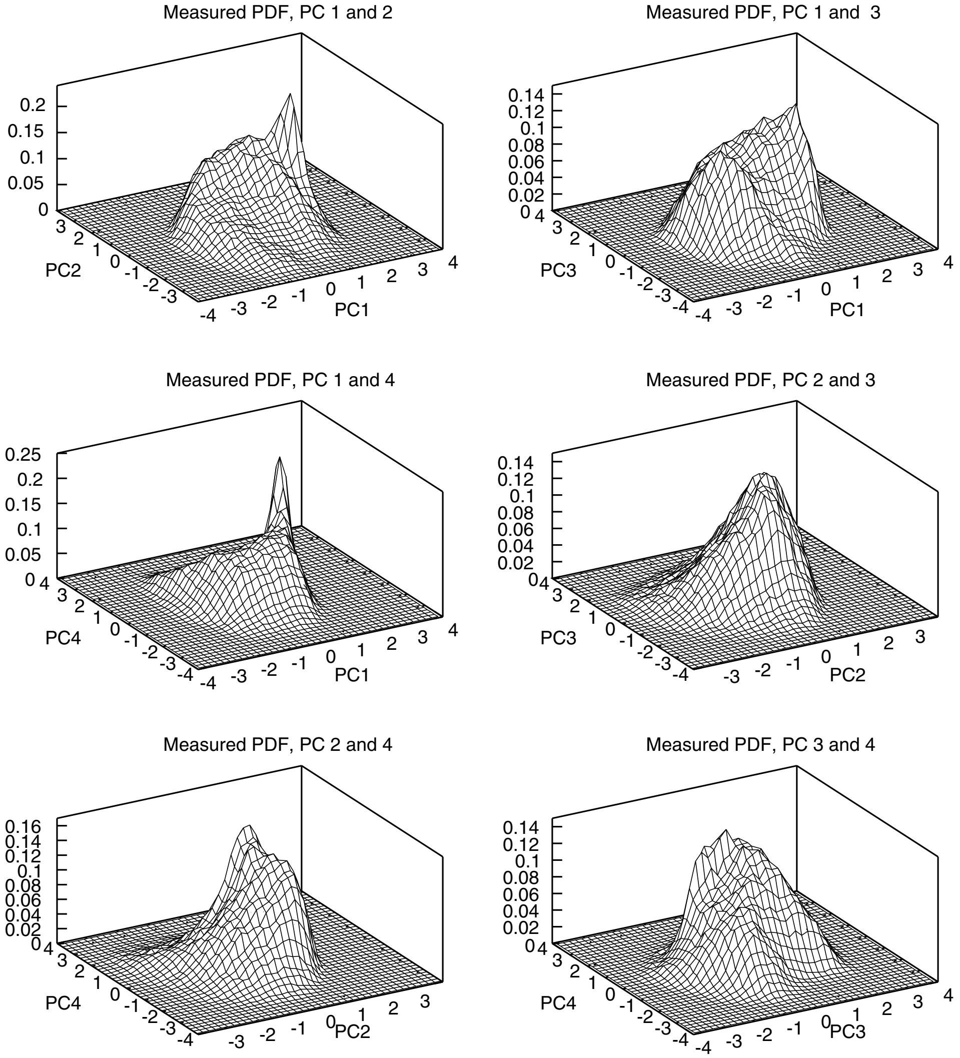


Fig. 1. The joint 2D PDFs of the PCs 1–4 as measured directly from the long-term model simulation through standard bin-counting.

Table 1
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 1 and 2, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	127	127	20.1	Basic vs.	–	0.467	0.245	0.179
BFGS	117	56	9.386	BFGS vs.	2.14	–	0.524	0.4
Scaled	32	32	4.919	Scaled vs.	4.09	1.91	–	0.731
S-BFGS	41	22	3.597	S-BFGS vs.	5.59	2.61	1.37	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 2
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 1 and 3, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	85	85	13.28	Basic vs.	–	0.296	0.264	0.0143
BFGS	54	23	3.929	BFGS vs.	3.38	–	0.894	0.05
Scaled	23	23	3.513	Scaled vs.	3.78	1.12	–	0.0541
S-BFGS	5	1	0.19	S-BFGS vs.	69.9	20.7	18.5	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 3
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 1 and 4, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	113	113	17.74	Basic vs.	–	0.418	0.291	0.217
BFGS	93	44	7.413	BFGS vs.	2.39	–	0.696	0.5
Scaled	32	32	5.16	Scaled vs.	3.44	1.44	–	0.746
S-BFGS	43	23	3.848	S-BFGS vs.	4.61	1.93	1.34	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 4
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 2 and 3, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	51	51	7.921	Basic vs.	–	0.359	0.35	0.14
BFGS	30	18	2.842	BFGS vs.	2.79	–	0.976	0.4
Scaled	18	18	2.774	Scaled vs.	2.86	1.02	–	0.399
S-BFGS	12	7	1.106	S-BFGS vs.	7.16	2.57	2.51	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 5
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 2 and 4, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	63	63	9.913	Basic vs.	–	0.365	0.327	0.262
BFGS	40	22	3.621	BFGS vs.	2.74	–	0.895	0.7
Scaled	21	21	3.241	Scaled vs.	3.06	1.12	–	0.802
S-BFGS	32	15	2.599	S-BFGS vs.	3.81	1.39	1.25	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 6
Comparison between different algorithms for the maximum entropy problem for the joint 2D PDF of PCs 3 and 4, moments of order up to 8

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	16	16	2.506	Basic vs.	–	0.846	0.945	0.474
BFGS	25	13	2.121	BFGS vs.	1.18	–	1.12	0.6
Scaled	15	15	2.367	Scaled vs.	1.06	0.896	–	0.502
S-BFGS	15	7	1.189	S-BFGS vs.	2.11	1.78	1.99	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 7
Comparison between different algorithms for the maximum entropy problem for the joint 3D PDF of PCs 1–3, moments of order up to 6

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	164	164	728.5	Basic vs.	–	0.868	0.262	0.168
BFGS	207	132	632.2	BFGS vs.	1.15	–	0.302	0.2
Scaled	40	40	191.1	Scaled vs.	3.81	3.31	–	0.639
S-BFGS	39	24	122.2	S-BFGS vs.	5.96	5.17	1.56	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

Table 8
Comparison between different algorithms for the maximum entropy problem for the joint 4D PDF of PCs 1–4, moments of order up to 4

Method	N_{iter}	N_{reort}	Time	Speedups	Basic	BFGS	Scaled	S-BFGS
Basic	24	24	1750	Basic vs.	–	0.789	1	0.718
BFGS	28	14	1380	BFGS vs.	1.27	–	1.27	0.9
Scaled	24	24	1750	Scaled vs.	1	0.788	–	0.718
S-BFGS	26	13	1257	S-BFGS vs.	1.39	1.1	1.39	–

Notations: N_{iter}, number of iterations; N_{reort}, number of polynomial reorthogonalizations; Time, wall clock time in seconds.

One can verify that setting α to the value in (28) automatically equates the new normalization constant and all corner moments of order $2M$ for the starting Gaussian distribution.

With the rescaling in (28), one can check what the difference between moments becomes for the example shown in (22). For the eighth order moment problem, we have

$$\begin{aligned} & & \alpha = 2.4784, & & \\ \mu_0 &= 1, & \mu_2 = 0.163, & & \mu_4 = 0.0795, \\ \mu_6 &= 0.0647, & \mu_8 = 0.0737, & & \mu_{10} = 0.108, \\ \mu_{12} &= 0.193, & \mu_{14} = 0.409, & & \mu_{16} = 1, \end{aligned} \tag{29}$$

that is, the 6-order difference in magnitudes of the moments is reduced to a single order. As we can see, the scaling in (28) is quite efficient, despite its apparent simplicity. After $\rho^*_\alpha(\vec{x})$ is found in the explicit form of (6), it is easy to convert it into $\rho^*(\vec{x})$ via (25).

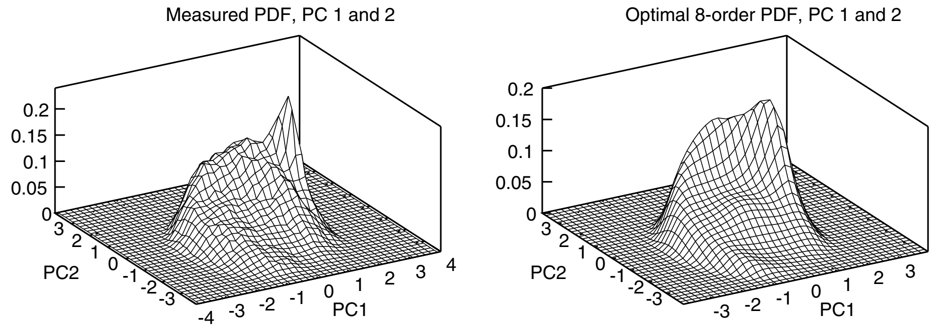


Fig. 2. The 2D probability density functions of PCs 1 and 2. Left picture – actual PDF recorded from model simulation by bin-counting, right picture – 8-order optimal PDF (44 total moment constraints).

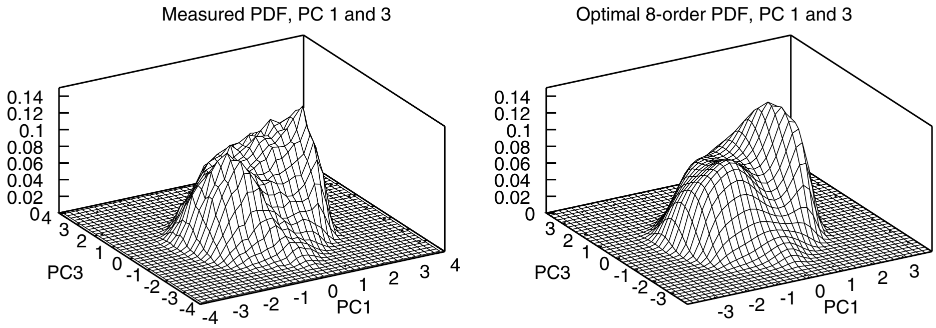


Fig. 3. The 2D probability density functions of PCs 1 and 3. Left picture – actual PDF recorded from model simulation by bin-counting, right picture – 8-order optimal PDF (44 total moment constraints).

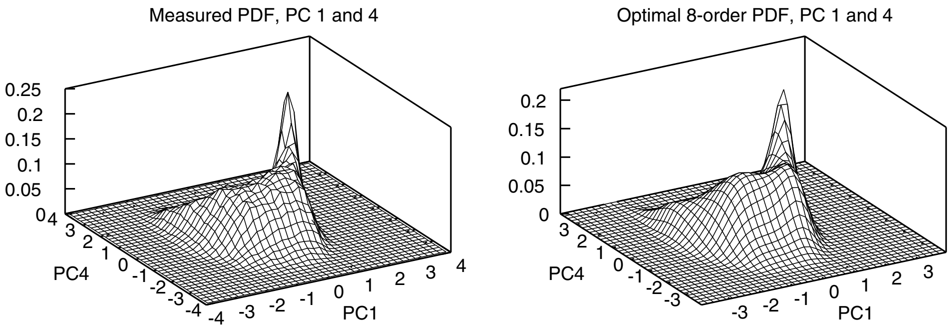
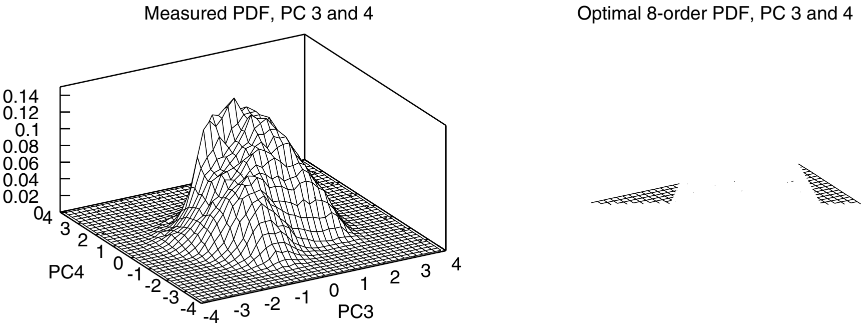


Fig. 4. The 2D probability density functions of PCs 1 and 4. Left picture – actual PDF recorded from model simulation by bin-counting, right picture – 8-order optimal PDF (44 total moment constraints).

4. Numerical testing

In Sections 3.1 and 3.2 we introduced two new modifications (namely, the BFGS iterations between polynomial reorthonormalizations and the new constraint scaling) to the basic moment-constrained maximum entropy algorithm, described above in Section 2 and developed earlier in [2]. These two modifications are not expected to significantly extend the ability of the basic algorithm to converge, but rather to reduce computational wall clock time in problems where the basic algorithm is already known to converge successfully. The two modifications can be used independently of each other, and as a result, produce four different versions of the algorithm when used in combinations; these versions are

- "Basic", the original algorithm from [2] where no modification is used.
- "BFGS", where the BFGS modification is used as described in Section 3.1.
- "Scaled", where the constraint scaling modification is used as described in Section 3.2.
- "S-BFGS", where both the BFGS and scaling modifications are used.

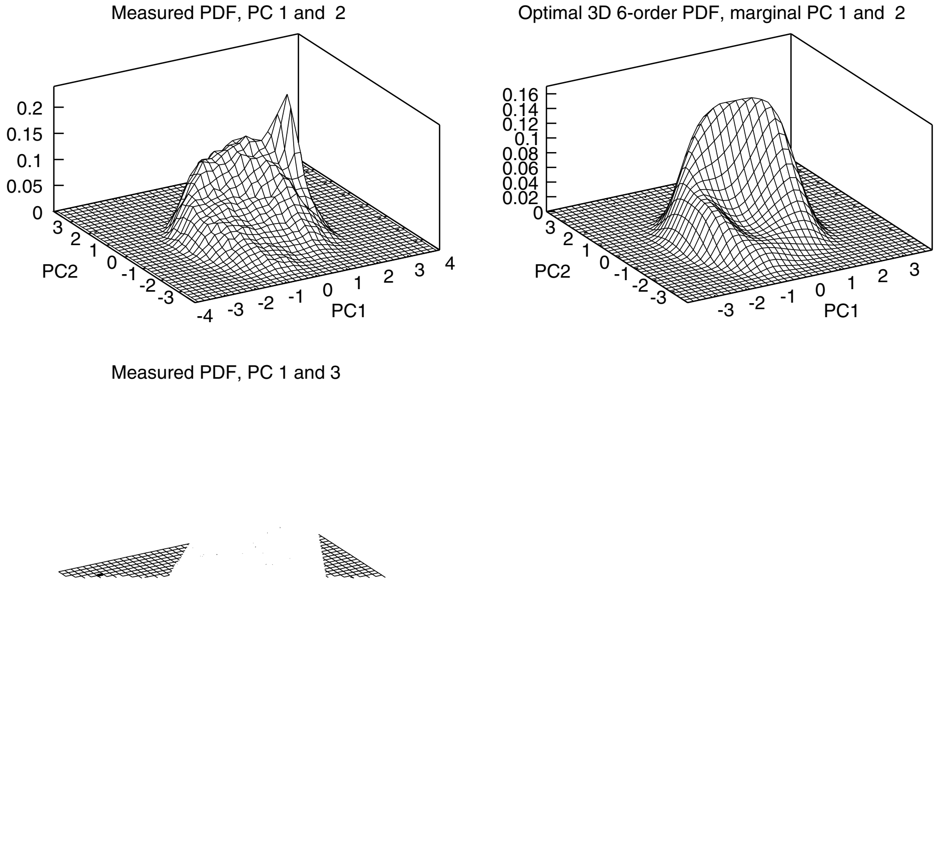


The test maximum entropy problems we consider here are the moment constrained climatological probability density functions of a model of wind stress-driven large-scale oceanic currents [18,19], which were earlier used in [2] to test the convergence of the basic maximum entropy algorithm. A subset of model variables, consisting of the first four principal components (PC) of the model is used to produce the 2-, 3- and 4-dimensional joint PDFs and their moment constraints. The joint 2-dimensional PDFs for PCs 1–4, as captured directly by bin-counting from the model simulation, are shown in Fig. 1. For more details on the model physics and simulation runs, see [2].

The basic algorithm, developed in [2], is known to converge for the probability distribution states, shown in Fig. 1, with the following sets of moment constraints: eighth order moments for 2D joint PDFs of PCs 1–4, 6-order moments for the 3D PDF of PCs 1–3 and 4-order moments for the 4D PDF of PCs 1–4. In the following sections we test convergence of the modified algorithms, described above, in this setting. The wall clock time, recorded for computations, corresponds to numerical simulations on an Athlon 64 3800+ processor. The speedup of one simulation relative to another is defined as the ratio of their wall clock times, which are given in seconds in Tables 1–8.

4.1. Maximum entropy problem of dimension 2 and moment order 8

In Figs. 2–7 and Tables 1–6 we show the results obtained for the two-dimensional PDFs of the PCs 1–4 (6 PDFs in total) with moment constraints of order up to 8 (44 moment constraints in total for each maximum entropy problem). Observe that the wall clock time, needed for convergence, drop significantly with either the BFGS or new constraint scaling applied, with generally better results obtained when both the BFGS and scaling are applied simultaneously. Speedups of between 2



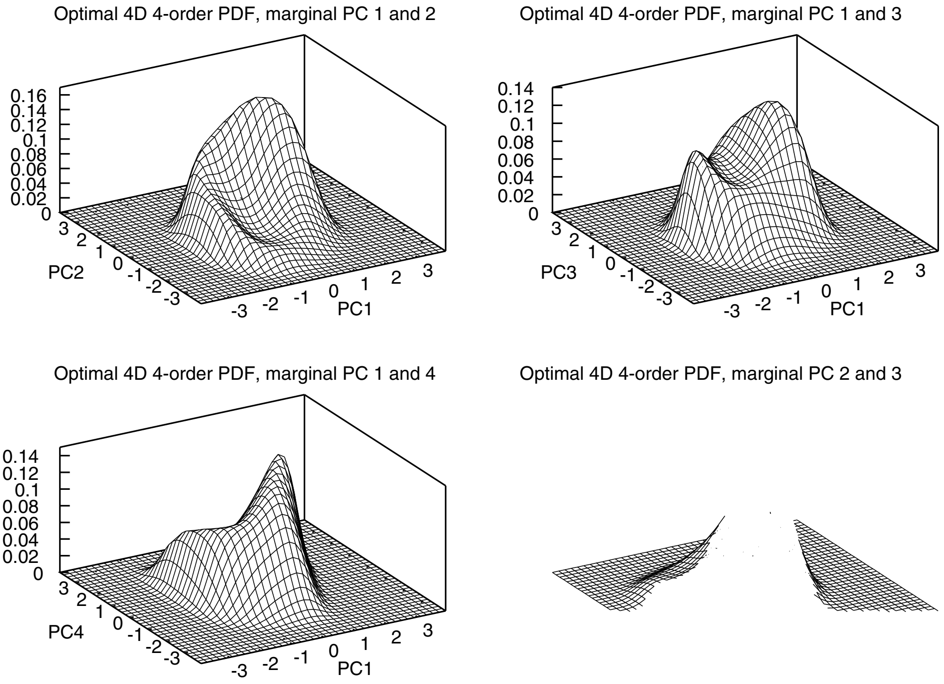
and 7 times relative to the basic algorithm are observed (excluding an apparently coincidental speedup of 70 for PCs 1 and 3). Note that using the BFGS method sometimes increases the total number of iterations, but substantially reduces the number of computationally expensive polynomial reorthogonalizations, thus still resulting in improved speedup. The new constraint scaling should be particularly efficient in the current setting, since it allows to reduce the difference in values of computed moments from six orders of magnitude to just one. Further, as we decrease the moment order for the maximum entropy problems of higher dimension, we will observe less impact on the speedup brought in by the constraint rescaling, since the difference in magnitudes of constraints generally diminishes with decreasing order.

4.2. Maximum entropy problem of dimension 3 and moment order 6

In Fig. 8 and Table 7 we show the results obtained for the three-dimensional PDF of the PCs 1–3 with moment constraints of order up to 6 (83 moment constraints in total). Again, observe that the wall clock time, needed for convergence, decreases with either BFGS or new constraint scaling applied, with the best result obtained when both the BFGS and scaling are applied simultaneously, with the speedup of six times obtained relative to the basic algorithm. The new constraint scaling here has a major impact on the wall clock time speedup, reducing the computational time by a factor of 3.5 both with or without BFGS, while the BFGS reduces the computational time by a factor of 1.2–1.5.

4.3. Maximum entropy problem of dimension 4 and moment order 4

In Fig. 9 and Table 8 we show the results obtained for the four-dimensional PDF of the PCs 1–4 with moment constraints of order up to 4 (69 moment constraints in total). Observe that the wall clock time, needed for convergence, moderately de-


Optimal 4D 4-order PDF, marginal PC 1 and 2

Optimal 4D 4-order PDF, marginal PC 1 and 3

Optimal 4D 4-order PDF, marginal PC 1 and 4

Optimal 4D 4-order PDF, marginal PC 2 and 3 creases with applying the BFGS iterations, yielding speedup of 1.27–1.39 relative to either basic or scaled algorithm without BFGS. On the other hand, the new constraint scaling has virtually no impact here. This is most likely due to the fact that the highest corner moment constraints, computed in this problem during the polynomial reorthogonalization, are of order 8, which for the Gaussian distribution of mean 0 and variance 1 are equal to 105. Thus, the unrescaled moment constraints differ in magnitude only by roughly two orders, thus rendering the constraint scaling largely unnecessary.

5. Summary

In the current work we develop and test two new improvements for the earlier developed algorithm [2] for the multidimensional moment-constrained maximum entropy problem. The first modification of the algorithm utilizes multiple Broyden–Fletcher–Goldfarb–Shanno (BFGS) iterations [6,7,9,13,21] to adaptively progress between points of polynomial reorthonormalization, as opposed to single Newton steps introduced originally in [2]. This strategy aims to decrease the total number of computationally expensive polynomial reorthonormalizations needed to achieve the optimal point, thus typically resulting in reduced computational wall clock time for the same problem. The second modification of the algorithm is a new moment constraint rescaling, designed to improve convergence by adjusting the magnitude of the highest-order corner constraints to the magnitude of the low-order constraints, thus reducing negative effects of the finite-precision machine arithmetic in the iterative convergence process. These two improvements are tested for the same set of multidimensional moment-constrained maximum entropy problems as in [2], for which the original algorithm from [2] is known to converge. This set of maximum entropy problems includes two-dimensional problems with moment constraints of order up to 8 (44 moment constraints in total), a three-dimensional problem with moment constraints of order up to 6 (83 moment constraints in total), and a four-dimensional problem with moment constraints of order up to 4 (69 moment constraints in total). The test maximum entropy problems are based on the long-term climatological statistics of a model for wind stress-driven large-scale ocean circulation [18,19]. It is found that the new improvements typically yield the speedup of 5–6 times as compared to the existing algorithm from [2] for the problems tested (also a coincidental anecdotal speedup of 70 was recorded). The greatest improvement in wall clock time speedup is obtained mainly in high (6–8) order maximum entropy problems, while in low-order four-dimensional test problem the observed speedup is rather moderate.

Future work in this direction will be aimed at the ability of the developed suite of algorithms to converge for problems with higher dimension and moment constraint order. As the number of iterations needed to converge for a higher-moment maximum entropy problem increases, as well as does the time to perform a single iteration in a higher-dimensional setting, various options for speeding up the iteration process will be studied, mainly focusing on parallel implementation of the Gauss–Hermite quadratures in a parallel computational environment. In particular, the use of contemporary graphic processing units (GPU), naturally suitable for SIMD floating point operations, will be considered for speeding up the quadrature computations.

Acknowledgments

The author is supported by the National Science Foundation Grant DMS-0608984 and the Office of Naval Research Grant N00014-06-1-0286.

References

[1] R. Abramov, A practical computational framework for the multidimensional moment-constrained maximum entropy principle, J. Comp. Phys. 211 (1) (2006) 198–209.
[2] R. Abramov, An improved algorithm for the multidimensional moment-constrained maximum entropy problem, J. Comp. Phys. 226 (2007) 621–644.
[3] R. Abramov, A. Majda, Quantifying uncertainty for non-Gaussian ensembles in complex systems, SIAM J. Sci. Comp. 26 (2) (2003) 411–447.
[4] R. Abramov, A. Majda, R. Kleeman, Information theory and predictability for low frequency variability, J. Atmos. Sci. 62 (1) (2005) 65–87.
[5] K. Bandyopadhyay, A. Bhattacharya, P. Biswas, D. Drabold, Maximum entropy and the problem of moments: A stable algorithm, Phys. Rev. E 71 (5) (2005).
[6] C. Broyden, The convergence of a class of double-rank minimization algorithms, J. Inst. Math. Appl. 6 (1970) 76–90.
[7] R. Byrd, P. Lu, J. Nocedal, A limited memory algorithm for bound-constrained optimization, SIAM J. Sci. Comp. 16 (6) (1995) 1190–1208.
[8] A. Dax, A modified Gram-Schmidt algorithm with iterative orthogonalization and column pivoting, Linear Algebra Appl. 310 (2000) 25–42.
[9] R. Fletcher, A new approach to variable metric algorithms, Comp. J. 13 (1970) 317–322.
[10] L. Giraud, J. Langou, When modified Gram-Schmidt generates a well-conditioned set of vectors, IMA J. Num. Anal. 22 (2002) 521–528.
[11] L. Giraud, J. Langou, M. Rozložník. On the round-off error analysis of the Gram-Schmidt algorithm with reorthogonalization. Technical Report TR/PA/02/33, CERFACS, 2002.
[12] L. Giraud, J. Langou, M. Rozložník. On the loss of orthogonality in the Gram-Schmidt orthogonalization process. Technical Report TR/PA/03/25, CERFACS, 2003.
[13] D. Goldfarb, A family of variable metric updates derived by variational means, Math. Comp. 24 (1970) 23–26.
[14] G. Golub, C. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[15] K. Haven, A. Majda, R. Abramov, Quantifying predictability through information theory: Small sample estimation in a non-Gaussian framework, J. Comp. Phys. 206 (2005) 334–362.
[16] R. Haydock, C. Nex, Comparison of quadrature and termination for estimating the density of states within the recursion method, J. Phys. C: Solid State Phys. 17 (1984) 4783–4789.
[17] R. Haydock, C. Nex, A general terminator for the recursion method, J. Phys. C: Solid State Phys. 18 (1985) 235–2248.
[18] J. McCalpin, The statistics and sensitivity of a double-gyre model: The reduced gravity, quasigeostrophic case, J. Phys. Oceanogr. 25 (1995) 806–824.
[19] J. McCalpin, D. Haidvogel, Phenomenology of the low-frequency variability in a reduced-gravity, quasigeostrophic double-gyre model, J. Phys. Oceanogr. 26 (1996) 739–752.

[20] D. Ormoneit, H. White, An efficient algorithm to compute maximum entropy densities, Econ. Rev. 18 (2) (1999) 127–140.
[21] D. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comp. 24 (1970) 647–656.
[22] I. Turek, A maximum-entropy approach to the density of states with the recursion method, J. Phys. C 21 (1988) 3251–3260.
[23] X. Wu, Calculation of maximum entropy densities with application to income distribution, J. Econ. 115 (2003) 347–354.
[24] Z. Wu, G. Phillips, R. Tapia, Y. Zhang, A fast Newton algorithm for entropy maximization in phase determination, SIAM Rev. 43 (4) (2001) 623–642.